



Universidade Federal do Ceará
Campus de Russas

**PROJETO PEDAGÓGICO DO CURSO
DE ENGENHARIA DE SOFTWARE
Grau: Bacharelado**

Novembro – 2013

PRESIDENTA DA REPÚBLICA
Dilma Vana Rousseff

MINISTRO DA EDUCAÇÃO
Aloizio Mercadante

UNIVERSIDADE FEDERAL DO CEARÁ

REITOR
Prof. Jesualdo Pereira Farias

VICE-REITOR
Prof. Henry de Holanda Campos

PRÓ-REITORA DE ADMINISTRAÇÃO
Profa. Denise Maria Moreira Chagas Correa

PRÓ-REITOR DE ASSUNTOS ESTUDANTIS
Prof. Ciro Nogueira Filho

PRÓ-REITORA DE EXTENSÃO
Profa. Márcia Maria Tavares Machado

PRÓ-REITOR DE GRADUAÇÃO
Prof. Custódio Luís Silva de Almeida

PRÓ-REITOR DE PESQUISA E PÓS-GRADUAÇÃO
Prof. Gil de Aquino Farias

PRÓ-REITOR DE PLANEJAMENTO
Prof. Ernesto da Silva Pitombeira

PRÓ-REITOR DE GESTÃO DE PESSOAS
Prof. Serafim Firmo de Souza Ferraz

ESTE PROJETO É PROPOSTO PELA PROGRAD/UFC COM BASE NO PROJETO PEDAGÓGICO DO CURSO DE ENGENHARIA DE SOFTWARE DO CAMPUS DA UFC EM QUIXADÁ, CURSO CRIADO EM 2009, DENTRO DO PROGRAMA DE APOIO A PLANOS DE REESTRUTURAÇÃO E EXPANSÃO DAS UNIVERSIDADES FEDERAIS (REUNI), CUJA EQUIPE RESPONSÁVEL FOI:

COORDENAÇÃO DO PROJETO

Davi Romero de Vasconcelos
Professor Adjunto do Campus de Quixadá
Flávio Rubens de Carvalho Sousa
Professor Assistente do Campus de Quixadá
Jefferson de Carvalho Silva
Professor Assistente do Campus de Quixadá
Lincoln Souza Rocha
Professor Assistente do Campus de Quixadá

Assessoria Pedagógica / PROGRAD

Custódio Luís Silva de Almeida
Pró-Reitor de Graduação
Inês Mamede
Coordenadora de Projetos e Acompanhamento Curricular
Yangla Kelly Oliveira Rodrigues
Diretora de Pesquisa e Desenvolvimento Curricular

SUMÁRIO

1. APRESENTAÇÃO	5
2. JUSTIFICATIVA	6
3. PRINCÍPIOS NORTEADORES	8
4. OBJETIVOS DO CURSO	9
5. PERFIL DO PROFISSIONAL A SER FORMADO (PERFIL DO EGRESSO)	10
6. COMPETÊNCIAS E HABILIDADES A SEREM DESENVOLVIDAS	11
7. ÁREAS DE ATUAÇÃO	13
8. METODOLOGIAS DE ENSINO E APRENDIZAGEM	14
9. ORGANIZAÇÃO CURRICULAR	15
9.1. UNIDADES CURRICULARES	17
9.2. DISCIPLINAS POR DEPARTAMENTO	20
9.3. EMENTÁRIO DAS DISCIPLINAS	20
9.4. ESTÁGIO SUPERVISIONADO	27
9.5. TRABALHO DE CONCLUSÃO DE CURSO	27
9.6. ATIVIDADES COMPLEMENTARES	28
10. INTEGRALIZAÇÃO CURRICULAR	28
11. ACOMPANHAMENTO E AVALIAÇÃO DO PROJETO PEDAGÓGICO	33
12. ACOMPANHAMENTO E AVALIAÇÃO DOS PROCESSOS DE ENSINO E DE APRENDIZAGEM	34

1. Apresentação

Este documento objetiva apresentar o projeto pedagógico do curso (PPC) de Engenharia de Software da Universidade Federal do Ceará (UFC), a ser criado no novo *Campus* no município de Russas, como parte da política nacional de expansão da educação superior.

Esta proposta de formação busca enfatizar o aspecto tecnológico no âmbito das técnicas computacionais, sem descuidar, naturalmente, do seu fundamento científico. Foram tomados como base para a construção deste projeto: o PPC do primeiro curso de Bacharelado em Engenharia de Software do Brasil, criado pelo Departamento de Informática da Universidade Federal de Goiás (UFG) e o PPC de Engenharia de Software da UFC/Quixadá; as diretrizes propostas pelas duas principais agremiações mundiais de profissionais da área de computação, ACM (*Association for Computing Machinery*) e IEEE *Computer Society*, no documento *Software Engineering Education Knowledge* (SEEK) [SEEK 2004] para a área de Engenharia de Software.

No Brasil até o presente momento existem 14 (quatorze) cursos de Bacharelado em Engenharia de Software, nas seguintes universidades¹: Universidade de Brasília (UNB); Universidade Federal do Amazonas (UFAM); Universidade da Região de Joinville (UNIVILLE); Universidade do Oeste de Santa Catarina (UNOESC); Pontifícia Universidade Católica de Minas Gerais (PUC-MINAS); Universidade Federal do Rio Grande do Norte (UFRN); Universidade Federal do Ceará (UFC) – *Campus* Quixadá; Universidade Federal de Goiás (UFG); Centro Universitário Salesiano de São Paulo (UNISAL); Centro Universitário UNIVATES (UNIVATES); Centro Universitário de Maringá (UNICESUMAR); Faculdade de Ensino Superior de São Miguel do Iguazu (FAESI); Universidade de Rio Verde (FESURV); Universidade Federal do Pampa (UNIPAMPA).

O curso de Engenharia de Software da UFC a ser implementado em Russas, configura-se como Bacharelado, na modalidade de ensino presencial, com oferta de 50 vagas anuais, com duração ideal de 8 semestres/4 anos, e máxima de 12 semestres/6 anos, e atividades previstas para os turnos da tarde e noite, caracterizando-se, portanto, como integral, haja vista que de acordo com a Portaria Normativa MEC nº. 40, publicada em 12 de dezembro de 2007, republicada em 2010, os cursos de graduação são de turno integral, caso a oferta de suas disciplinas e atividades sejam inteira ou parcialmente em mais de um turno (manhã e tarde, manhã e noite, ou tarde e noite) exigindo a disponibilidade do estudante por mais de 6 horas diárias durante a maior parte da semana.

O maior desafio assumido por este curso é articular teoria e prática, de modo a formar um profissional apto a atuar na indústria de software e conscientizar o egresso quanto à necessidade de manter-se continuamente atualizado diante do progresso incessante que é uma característica dessa área de atuação. Objetiva-se formar um profissional empreendedor, capaz de lidar com técnicas avançadas de gerenciamento de projetos, qualidade de processos e produtos e inovação tecnológica. Para tanto, propõe-

¹ Fonte: www.emec.mec.gov.br

se um modelo pedagógico capaz de adaptar-se à dinâmica das demandas da sociedade, em que a graduação passa a constituir-se numa etapa de formação inicial em processo de educação permanente.

O PPC de Engenharia de Software é encarado como um moderno instrumento de gestão, que busca extrapolar a simples confecção de um documento para se caracterizar como um processo dinâmico de ação-reflexão-ação, objetivando uma permanente adequação do ensino superior das Tecnologias de Informação e Comunicação (TIC) às práticas e exigências do mercado de trabalho, pautando-se na necessidade da formação ética e cidadã desses profissionais.

2. Justificativa

A exigência de responsabilidade e competência, atributos subjacentes à proposta de um curso superior, é pressuposto inelutável, pois cabe à universidade responder às pressões emergentes no que tange ao florescimento de diferentes ramos das ciências, da tecnologia e das humanidades; isto impõe à Instituição o diálogo com a sociedade, na busca de satisfazer à demanda por formação de profissionais e de sujeitos conscientizados do seu papel transformador da realidade.

Os avanços da Tecnologia da Informação e Comunicação (TIC) têm impulsionado uma demanda pelo desenvolvimento de sistemas de software mais complexos, confiáveis e de qualidade. Neste cenário, surge a necessidade de profissionais qualificados para esta atividade, aqui denominados *Engenheiros de Software*. Além disso, a dificuldade de produzir software dentro dos prazos estipulados, com o orçamento previsto e satisfazer os requisitos dos clientes representa outro grande desafio para área.

A Engenharia de Software é “a aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de software” [IEEE 1990]. Suas principais bases estão na Ciência da Computação e na Matemática [SEEK 2004] e se dedicam aos problemas práticos da produção de software [Sommerville 2006]. O conjunto de conhecimentos pertinentes a Engenharia de Software é documentado em SWEBOK [2004]. A Engenharia de Software usa a matemática, a ciência da computação e a sistemática das engenharias para resolver problemas em domínios de aplicação.

A sociedade para a promoção do software brasileiro, Associação para Promoção da Excelência do Software Brasileiro [Softex], e órgãos de fomento à pesquisa como o Conselho Nacional de Desenvolvimento Científico e Tecnológico [CNPq] e a Financiadora de Estudos e Projetos [FINEP] têm estimulado o desenvolvimento de software pelo caráter estratégico para o país por meio de editais específicos.

Vale salientar que o Estado do Ceará é reconhecido como um dos maiores pólos de desenvolvimento de software do país e conta com mais de 200 empresas atuando na área. O sucesso das leis de incentivo a empresas de base tecnológica instaladas na Região Nordeste demonstra a competência de empresas locais no cenário atual. No entanto, a formação de profissionais qualificados contribuirá para a autossustentabilidade das empresas após a retirada dos incentivos.

Desde 1975, a UFC vem oferecendo cursos de graduação na área de Computação, tendo iniciado com a formação de Tecnólogo em Processamento de Dados, posteriormente transformado em Bacharelado em Computação (1985). O Departamento de Computação da UFC realizou esforços de formação de professores doutores e, em 1995 inaugurou a pós-graduação *stricto sensu* com o Mestrado em Ciência da Computação. Dez anos mais tarde foi criado o Doutorado em Ciência da Computação.

Em 2006, foi criado o curso de Engenharia da Computação (Bacharelado) no município de Sobral, como parte do primeiro movimento de expansão da UFC para o interior do Estado do Ceará.

Através do Programa REUNI, foram criados, respectivamente, nos anos de 2007, 2009 e 2012, os cursos de Sistemas de Informação, de Engenharia de Software e de Ciência da Computação, Bacharelados e localizados no *Campus* de Quixadá.

O processo de expansão da UFC na direção do interior do estado vem responder às antigas demandas da sociedade. Por outro lado, o Estado do Ceará conta, há 58 anos, com a contribuição desta universidade na formação de pessoal altamente qualificado, na geração e preservação de conhecimento, na inovação tecnológica e na integração com a sociedade através de atividades e projetos de extensão.

O Estado do Ceará tem apresentado uma impressionante aceleração de sua urbanização nas duas últimas décadas. O significativo crescimento dos setores produtores de bens-salário tem incentivado a expansão, ainda que tímida, dos setores de bens intermediários e bens de capital no estado. O conhecimento tornou-se o principal recurso econômico. Na sociedade do conhecimento, e principalmente na área tecnológica, esse ativo se torna rapidamente obsoleto, obrigando os profissionais a realizar reciclagens periódicas. Nos últimos anos, a competitividade incentivada principalmente pela globalização exigiu uma reformulação das empresas e principalmente no perfil dos profissionais especializados em informática. Em termos de economia, os resultados mostram que o Ceará vem crescendo a taxas maiores que o Nordeste e Brasil.

Todos esses fatores que envolvem uma demanda crescente no Estado do Ceará por profissionais da área de TIC, atrelados a flexibilidade curricular dos programas de ensino em todos os níveis e a demanda crescente de serviços educacionais da sociedade cearense estimulam a criação do curso de Engenharia de Software no município de Russas. Este se propõe a oferecer ao seu corpo discente uma formação universitária compatível com as peculiares questões relacionadas às TIC e a indústria de software no Brasil. Desta forma, a UFC, conhecedora da função social do profissional atuante na área de TIC e no mercado de software, busca a formação de um profissional especializado e integrado à realidade social onde está inserido.

O município de Russas² está localizado na região do Baixo Jaguaribe. Situa-se a 165 km da capital Fortaleza, tendo como principal acesso a BR 116. A cidade constitui um dos mais importantes centros populacionais e econômicos do Vale do Jaguaribe. É

² Fonte das Informações sobre o Município de Russas: www.russas.ce.gov.br

conhecida como a “Capital do Vale do Jaguaribe”, “Terra da Laranja Doce”, “Terra das Telhas Vermelhas”, e “Terra de Dom Lino”.

Russas é um importante pólo econômico do Vale do Jaguaribe e do Estado do Ceará. Localizada numa região de solo fértil do Vale Jaguaribano, sempre foi ponto estratégico para o transporte de pessoas e mercadorias, por ali ter passado a Estrada Real do Jaguaribe no período colonial, depois a estrada Transnordestina e hoje a BR 116. Após o ciclo da Carne de Charque, a economia do município passou pelo ciclo do algodão, o ciclo da carnaúba, e o da laranja. Esta última lhe rendeu o título de “Terra da Laranja Doce”, pois se criou uma verdadeira marca para a laranja da região, a “Laranja de Russas” conhecida nacionalmente.

Nos dias de hoje a economia russana é baseada em movimentado Comércio, prestação de serviços, o agronegócio, bem como as Indústrias (calçadista, cerâmicas, peças automotivas, etc). O PIB do município foi de R\$ 431.695,00 reais em 2008, de acordo com o IBGE. Graças ao bom desempenho no quesito emprego e renda, Russas ficou entre os 10 municípios de Estado do Ceará com melhor IFDM (Índice FIRJAN de Desenvolvimento Municipal).

Os indicadores positivos do município de Russas, o tornam atrativo para o desenvolvimento de novos empreendimentos, incluindo aqueles relacionados à prestação de serviço e desenvolvimento industrial. Neste último segmento, já há alguns anos, a cidade é sede de filial da indústria Dakota Calçados, uma das maiores empresas calçadistas da América Latina, que constitui o maior empregador da cidade, gerando em torno de 4.000 empregos diretos. Desde sua instalação, em 1997, a Dakota tornou-se vetor de desenvolvimento para novas indústrias na região jaguaribana.

3. Princípios Norteadores

O presente projeto é destacado como um empenho político-pedagógico que visa uma formação em Engenharia de Software socialmente consciente e instigante, ultrapassando limites disciplinares e considerando o saber como uma construção social. Essa vertente orientadora reafirma como elementos fundantes, para atuar como profissional da Computação, princípios da ética democrática, dos quais se destaca: a dignidade humana, a justiça, o respeito mútuo, a participação, a responsabilidade, o diálogo e a solidariedade, na sua atuação como profissional e como cidadão.

Nesse sentido, o curso de Engenharia de Software defende os seguintes princípios norteadores:

- O ser humano seja o princípio e fim de todo processo formativo no qual haja comprometimento com a ética na busca da verdade e do conhecimento;
- A liberdade do pensamento e expressão;
- O compromisso com o fortalecimento da cultura acadêmica, através da integração entre ensino, pesquisa e extensão;
- A ação-reflexão-ação e a articulação entre teoria e prática, técnica e humanismo;

- A capacidade de adaptação à evolução tecnológica.
- A articulação entre a formação básica, profissional e humanista.

Como os demais cursos criados no *Campus* de Russas, em termos curriculares, este projeto pedagógico assume como preocupação central, a formulação de uma proposta que seja capaz de formar cidadãos e profissionais que não se limitem apenas ao exercício técnico-profissional, mas que estejam aptos a uma atuação ético-política, comprometida com as transformações qualitativas do mundo em que vivemos, na perspectiva da promoção de uma sociedade democrática, plural e justa.

As orientações curriculares visam, nestes termos: à promoção de uma cultura acadêmica, de caráter não tecnicista, pautada na ampliação dos espaços de aprendizagem, na diversidade e integração crescente dos conteúdos científicos e artísticos e na capacidade de lidar com sua intensa mutação na sociedade contemporânea.

Sob esta ótica, defende-se como princípios curriculares:

- Flexibilização Curricular – recomenda-se a presença de uma proporção significativa de conteúdos de natureza optativa nos currículos e a redução das exigências de pré-requisitos, sempre que pertinente, de modo a permitir que o aluno participe do processo de definição do seu percurso acadêmico.

- Caráter inter e transdisciplinar – compreendido como a valorização da articulação dos diversos campos do saber, vinculando a formação técnica à formação humanística, promovendo a relação teoria/técnica/prática articulada a uma dimensão ético-estética, e dando ao currículo uma perspectiva de conjunto, que favoreça a superação da visão fragmentada do conhecimento.

- Atualização Permanente – recomenda-se que o PPC fomente o processo de atualização permanente, que permita estabelecer o aprimoramento e/ou a correção de trajetórias, a incorporação dos avanços científicos e tecnológicos, as inovações artísticas e as conquistas substantivas nos diversos campos do conhecimento.

4. Objetivos do Curso

O Curso de Bacharelado em Engenharia de Software da UFC em Russas tem como objetivo formar profissionais aptos a introduzirem melhorias e a participarem efetivamente de empreendimentos de software voltado para os mercados local e global, oferecendo a base teórica suficiente para que os seus egressos possam manter-se constantemente atualizados. Além disso, é objetivo deste curso preparar profissionais para construir, usando as técnicas da Engenharia de Software, sistemas de software corretos, completos, seguros, amigáveis, usáveis, com qualidade, fáceis de manter e de baixo custo.

5. Perfil do Profissional a ser Formado (Perfil do Egresso)

As organizações contemporâneas têm na tecnologia da informação um elemento estratégico, na medida em que as soluções tecnológicas automatizam processos organizacionais e são fonte de vantagens competitivas através da análise de cenários, apoio ao processo decisório e definição e implementação de novas estratégias organizacionais. Assim, cresce a preocupação com a coleta, armazenamento, processamento e transmissão da informação na medida em que a disponibilidade da informação correta, no momento apropriado, é requisito fundamental para a melhoria contínua da qualidade e competitividade organizacionais, o que implica em considerar a crescente relevância da Engenharia de Software no desenvolvimento de sistemas de software com eficiência, qualidade e a baixo custo.

O egresso do curso de Engenharia de Software estará apto a atuar na indústria de desenvolvimento de software. O(a) bacharel(a) em Engenharia de Software deve ser capaz de efetivamente contribuir com equipes na produção de modelos abstratos correspondentes a software e realizá-los por meio de código funcionando em contexto real.

Da perspectiva pessoal o egresso deve ser capaz de:

- Trabalhar de forma harmoniosa e efetivamente auxiliar na elaboração de produtos atribuídos a equipes;
- Valorizar e iniciar longo processo de formação de sua própria reputação na área;
- Desenvolver atitudes e posturas ativas.

Da perspectiva cognitiva o egresso deve ser capaz de:

- Elicitar, analisar, modelar, especificar (documentar), validar e gerenciar requisitos de software;
- Projetar (design) software (arquitetura e projeto detalhado), incluindo modelagem, análise e avaliação da qualidade, princípios, estilos, métodos, modelos arquiteturais e padrões de projeto;
- Construir (programar) software com qualidade e em equipe, incluindo métodos, técnicas, tecnologias e ferramentas;
- Realizar atividades de manutenção de software;
- Planejar e executar atividades pertinentes à qualidade de software, incluindo verificação, validação, revisões, inspeções e testes;
- Gerenciar pequenos projetos de desenvolvimento de software, incluindo estimativas de custo de software;
- Personalizar processos de software em conformidade com modelos de melhoria de processos;

- Transmitir ideias com clareza (seja na forma verbal ou escrita).

Da perspectiva tecnológica e pragmática o egresso deve ser capaz de:

- Exercitar o conhecimento, empregando tecnologias e ferramentas para desenvolvimento de software complexo por meio da participação em equipes de projeto;
- Selecionar tecnologias apropriadas para um dado contexto.

6. Competências e Habilidades a serem Desenvolvidas

O desempenho das atividades inerentes às áreas de atuação em Engenharia de Software, aqui agrupadas em três pilares de formação, exige uma ação profissional fundamentada no conhecimento teórico-prático aprofundado da aplicação das soluções tecnológicas oferecidas pela ciência da computação a problemas existentes no mercado de software.

Para a formação do perfil do egresso pretendida pelo curso de Engenharia de Software, faz-se necessário o desenvolvimento de determinadas competências e habilidades alinhadas aos conhecimentos técnico-científicos adquiridos ao longo do curso.

Ao lado disso, o presente projeto prevê a ação integradora em disciplinas de outras áreas, visando fortalecer a interdisciplinaridade para a formação do futuro profissional.

Inicialmente esta exigência implica em uma capacitação profissional que integre **formação humanística; formação técnica;** e a busca incessante pela **interdisciplinaridade**. Além disso, a capacitação deve incluir o desenvolvimento de habilidades de relacionamento interpessoal, comunicação e trabalho em equipe, na medida em que são características cada vez mais importantes na atuação profissional. Assim, o profissional de Engenharia de Software deve dispor de uma sólida formação conceitual (conhecimento explícito) aliada a uma capacidade de aplicação destes conhecimentos científicos em sua área de atuação (conhecimento tácito) de forma a agregar valor econômico à organização e valor social ao indivíduo, bem como realize seu trabalho de modo ético e consciente da sua realidade econômica, política e social.

Formação Humanística:

Esse elemento é imprescindível para a formação da postura profissional do egresso e é abordado em uma disciplina própria, dedicada especificamente ao assunto, e estimulado nas demais. Dessa forma, é esperado que o profissional de Engenharia de Software tenha as seguintes competências:

- a) Ser criativo e inovador na proposição de soluções para os problemas e oportunidades identificados nas organizações;
- b) Saber expressar-se nas formas oral e escrita de forma clara, empregando técnicas de comunicação apropriadas para cada situação;

- c) Saber trabalhar em equipe, somando as suas competências àquelas de profissionais de outras áreas, em prol do desenvolvimento de projetos com objetivos comuns;
- d) Ter uma visão contextualizada da área de Engenharia de Software em termos políticos, sociais, econômicos e ambientais;
- e) Atuar social e profissionalmente de forma ética e cidadã.

Formação Técnica:

Tem como base as recomendações do SWEBOK [SWEBOK] e do SEEK [SEEK 2004]. Adicionalmente, no que tange o gerenciamento de projetos, são observadas as recomendações contidas no PMBOK [PMBOK]. Dessa forma, é esperado que o profissional de Engenharia de Software tenha as seguintes competências:

- a) Compreender a dinâmica empresarial decorrente de mercados mais exigentes e conscientes de seus direitos e das novas necessidades sociais, ambientais e econômicas;
- b) Participar do desenvolvimento e implantação de novos modelos de competitividade e produtividade nas organizações no que tange à produção de sistemas de software;
- c) Diagnosticar e mapear, com base científica, problemas e pontos de melhoria nas organizações, propondo alternativas de soluções baseadas nos fundamentos da Engenharia de Software;
- d) Elicitar, especificar, rastrear e gerenciar requisitos de software, bem como projetar, especificar, implementar, implantar, verificar, validar e manter (evoluir ou corrigir) sistemas de software de maneira apropriada;
- e) Participar da implantação e monitoramento dos processos organizacionais de empresas de software, identificando as possíveis mudanças que podem surgir em função dos objetivos das organizações, das exigências do mercado, da legislação vigente e dos avanços nos modelos de qualidade de processo e produto no âmbito da Engenharia de Software;
- f) Criar, implantar, refinar e avaliar processos organizacionais, relacionados à construção de software, que visam garantir maior produtividade, qualidade, alinhamento da produção aos objetivos das organizações e conformidade com a legislação vigente nas esferas local, regional, nacional e mundial;
- g) Planejar e gerenciar projetos de software observando as necessidades dos clientes, a tríade tempo, escopo e custo, o modelo de processo adotado na organização e a legislação vigente.

Interdisciplinaridade:

Construir sistemas de software implica, necessariamente, no emprego de dois domínios: 1) o de Engenharia de Software ligado à computação e; 2) o domínio onde está inserido o problema que motiva a construção do software. Dessa forma, é esperado que o profissional de Engenharia de Software desenvolva as seguintes competências:

- a) Deve ser capaz de trabalhar em equipe multidisciplinar;
- b) Deve possuir uma visão holística do mundo, da sociedade e de suas dinâmicas;

- c) Ser capaz de investigar e entender problemas em diversificados domínios de aplicação;
- d) Capacidade de autoaprendizagem no que tange ao entendimento de outros domínios do conhecimento.

Além disso, na formação profissional do aluno, busca-se orientá-lo no sentido de adaptar-se às mudanças e novidades da área das TIC. Para tanto, torna-se indispensável o desenvolvimento da inteligência emocional, tendo autoconhecimento, automotivação, sociabilidade e liderança como algumas características a serem desenvolvidas durante a formação do aluno através do incentivo de atividades em equipe.

7. Áreas de Atuação

O egresso terá condições de assumir um papel de agente transformador do mercado, sendo capaz de provocar mudanças através da incorporação de novas tecnologias da informação na solução dos problemas e propiciando novos tipos de atividades, agregando:

- a) Domínio de novas tecnologias da informação e gestão da área de Engenharia de Software, visando melhores condições de trabalho e de vida;
- b) Conhecimento e emprego de modelos associados ao uso das novas tecnologias da informação e ferramentas que representem o estado da arte na área;
- c) Conhecimento e emprego de modelos associados ao diagnóstico, planejamento, implementação e avaliação de projetos de sistemas de software aplicados nas organizações;
- d) Uma visão humanística consistente e crítica do impacto de sua atuação profissional na sociedade e nas organizações.

Desta forma, não exclusivamente, o egresso do curso poderá atuar como:

- Analista de sistemas de software
- Desenvolvedor de sistemas de software
- Gerente/Analista de configuração
- Projetista de sistemas de software
- Arquiteto de software
- Gerente/Analista de qualidade de software
- Gerente/Analista de teste de software
- Gerente de projetos de software
- Consultor/Auditor de sistemas software
- Professor e/ou Pesquisador

8. Metodologias de Ensino e Aprendizagem

O curso de Engenharia de Software compreende a importância, e por isso, pretende desenvolver metodologias de ensino e aprendizagem inovadoras, que façam uso de novas tecnologias, baseiem-se na construção do conhecimento e não na sua mera transmissão, instituindo novos papéis para professores e alunos.

Nesse sentido, será incentivada e valorizada a atuação do docente como mediador/facilitador dos processos de ensino e de aprendizagem, intermediando a relação entre os discentes e o conhecimento. Também se espera que os professores procurem atuar de forma comprometida com o PPC, procurando sempre que possível relacionar os assuntos abordados em sua disciplina com outros assuntos, permitindo ao estudante ter a visão holística do curso.

A metodologia de ensino poderá incluir aulas expositivas, dialogadas e práticas, assim como o uso de diferentes recursos audiovisuais, visitas técnicas, mini-cursos e palestras. Intercâmbios institucionais também serão recursos importantes e relevantes na formação do graduando. A preocupação com a articulação entre a teoria e prática e a interdisciplinaridade deverá permear a implementação do currículo do curso, cabendo aos professores encontrar estratégias para sua promoção.

Quanto aos alunos será estimulada a sua participação ativa, como seres pensantes e coparticipes da sua formação. Nessa direção, este curso defende como um de seus princípios, a flexibilidade curricular, conforme já citado anteriormente. A flexibilidade ou flexibilização curricular amplia o leque de oferta de componentes de livre-escolha dos alunos (disciplinas optativas, optativas-livres e atividades complementares) de modo que o estudante tenha maior possibilidade de construção do seu percurso formativo, valorizando e incentivando a sua autonomia.

As metodologias de ensino e de aprendizagem terão como referencial o desenvolvimento das competências e habilidades definidas neste curso, tendo em vista o perfil do profissional delineado.

Considerando essas premissas e as especificidades de um curso da área das TIC, que requer o contato contínuo com os recursos computacionais, o conhecimento básico em matemática e a produção de novos saberes, processos, métodos e instrumentos, o curso de Engenharia de Software do Campus de Russas, procurará levar a cabo as seguintes estratégias pedagógicas:

I) Uso Intensivo de Laboratórios:

Um curso de Engenharia de Software se caracteriza por uma intensa interação entre hardware e software, sendo, portanto, importante o incentivo ao desenvolvimento de atividades nos laboratórios específicos do curso, notadamente de desenvolvimento de software. Note-se que privilegiamos uma formação que transcenda a sala de aula e que privilegie a interação entre o prático e o teórico, reforçando uma vez mais o papel dos laboratórios e da biblioteca como elementos centrais de qualidade do curso.

II) Atividades de Nivelamento :

O curso demanda conhecimentos prévios de matemática do ensino médio. Alguns alunos nos primeiros semestres do curso necessitarão de reforço escolar em matérias ligadas à matemática. Pretende-se aqui fornecer atividades extracurriculares para o ensino da matemática do ensino médio.

III) Atividades Conjuntas da Graduação e Pós-graduação :

O Departamento de Computação e o Departamento de Engenharia de Teleinformática da UFC em Fortaleza poderão contribuir fazendo com que suas atividades de pós-graduação, *stricto e lato sensu*, interajam com o curso de Engenharia de Software em Russas.

- Oferta de seminários de pesquisa abertos à participação de professores e alunos de Russas;
- Criação de projetos e grupos de pesquisa envolvendo as duas unidades;
- Oferta de palestras em Fortaleza e em Russas;
- Oportunidade de qualificação acadêmica para professores e alunos em seus cursos de pós-graduação.

IV) Mobilidade Acadêmica

O MEC introduziu o programa de *Mobilidade Acadêmica* que permite o intercâmbio entre alunos de Instituições Federais de Ensino Superior (IFES). Este programa se mostra amplamente adequado para que alunos do curso possam, por períodos determinados, conhecer a realidade da formação de outros cursos, ampliando suas possibilidades de formação.

9. Organização Curricular

A proposta de formatação do curso foi elaborada a partir das necessidades regionais para formar profissionais aptos a atuarem nas diversas áreas relacionadas à informática no Estado do Ceará, advindas da expansão do mercado de TIC. Através de parcerias mantidas com empresas e instituições que trabalham o desenvolvimento da informática no estado, como o Instituto do Software do Estado do Ceará, o projeto do curso procurará priorizar o atendimento às demandas dos cidadãos, da sociedade e do mercado de trabalho.

O currículo do curso oferece flexibilidade ao utilizar o critério de pré-requisitos mínimos para as disciplinas, o que facilita para o aluno uma melhor oferta de disciplinas no curso e oferta de atividades complementares. As parcerias com empresas da área de Informática do estado permitirão que a coordenação do curso e os professores tenham uma avaliação permanente da demanda local, e com isso uma informação que possibilita uma atualização constante do curso e seu currículo.

Toda a estrutura curricular do curso de Bacharelado em Engenharia de Software foi elaborada de forma a contemplar os objetivos do curso e atingir o perfil profissional proposto. A organização do currículo permite a compreensão, o entendimento e o conhecimento para aplicar e desenvolver modelos, utilizando as novas tecnologias e metodologias, assegurando as inter-relações com outras áreas do conhecimento, contribuindo assim, com o processo de compreensão e transformação da realidade, desenvolvendo no discente não só competências, como também formando um cidadão consciente do seu papel na sociedade e alicerçado nos princípios da ética e da cidadania.

A matriz curricular compreenderá uma formação de 3.072 horas (equivalendo a 186 créditos³) com tempo ideal para conclusão do curso estimado em 4 (quatro) anos ou 8 (oito) semestres letivos. Sendo assim, o estudante do Curso de Engenharia de Software, grau Bacharelado, deverá observar o tempo máximo para a sua conclusão, estipulado em 6 (seis) anos ou 12 (doze) períodos letivos. Os alunos deverão cursar uma carga horária mínima por semestre de 4 créditos (64 horas) e, no máximo, 32 créditos (512 horas).

As disciplinas do Curso de Engenharia de Software serão de três tipos: obrigatórias, optativas (da integralização curricular do curso) e optativas-livres (de fora da integralização curricular do curso). Além das disciplinas, o currículo compreenderá algumas atividades extraclasse: estágio supervisionado, trabalho de conclusão de curso e atividades complementares. Estas últimas abrangerão atividades de extensão, monitoria, iniciação científica, participação e organização de seminários e palestras, entre outras regulamentadas pela Resolução CEPE/UFC n.º 7/2006.

Devido ao rápido surgimento de novas tecnologias de informação, faz-se necessário a inclusão de disciplinas optativas que abordem tais temas. Para tanto, estão previstas algumas disciplinas chamadas Tópicos Especiais que serão ofertadas de acordo com a demanda por estas tecnologias.

A carga horária do curso terá a seguinte distribuição:

Componentes	Nº de Horas	Nº de Créditos	Percentual sobre a carga horária total do curso
Disciplinas Obrigatórias	1.728	108	56,25%
Disciplinas Optativas	320	20	10,42%
Disciplinas Optativas-Livres	256	16	8,33%
Estágio Supervisionado	320	20	10,42%
Trabalho Final de Curso	160	10	5,21%
Atividades Complementares	288	18	9,37%
Total	3.072	192	100%

9.1. Unidades Curriculares

³ Na UFC um crédito é equivalente a 16 horas-aula.

As tabelas a seguir apresentam os conteúdos a serem observados na construção de projetos pedagógicos de cursos de Bacharelado em Engenharia de Software. Cada um deles poderá ser operacionalizado através de uma ou mais disciplinas e atividades em um currículo específico a ser implementado.

I. Formação Básica

A área de formação básica é composta pela Formação Básica em Ciência da Computação e Formação Básica em Matemática. A Formação Básica em Engenharia de Software foi adicionada em virtude do objetivo do curso.

I.1 Formação Básica em Ciência da Computação

Conteúdo	Disciplina
Programação	Fundamentos de Programação
	Programação Orientada a Objetos
	Estruturas de Dados
	Linguagens de Programação
Teoria da Computação e Algoritmos	Linguagens Formais
	Projeto e Análise de Algoritmos
	Teoria da Computação
Arquitetura de Computadores	Arquitetura de Computadores

I.2 Formação Básica em Matemática

Conteúdo	Disciplina
Matemática	Matemática Básica
	Matemática Discreta
	Lógica para Computação
	Probabilidade e Estatística

I.3 Formação Básica em Engenharia de Software

Conteúdo	Disciplina
Introdução a Engenharia de Software	Introdução à Ciência da Computação e Engenharia de Software
	Introdução a Processos e Requisitos de Software

II. Formação Tecnológica

A área de formação tecnológica é composta por um conjunto de conteúdos relacionados à Engenharia de Software e às Tecnologias da Informação e Comunicação utilizadas no desenvolvimento de sistemas de software.

Conteúdo	Disciplina
Engenharia de Software	Modelagem e Análise de Software
	Projeto Detalhado de Software
	Interface Humano-Computador
	Gerência de Projetos de Software
	Processo de Software
	Requisitos de Software
	Qualidade de Software

	Reuso de Software
	Arquitetura de Software
	Verificação e Validação
Compiladores	Compiladores
Banco de Dados	Fundamentos de Banco de Dados
Inteligência Artificial	Inteligência Artificial
Engenharia de Software Aplicada	Integração de Aplicações
	Especificação Formal de Software
	Gerência de Configuração
	Manutenção de Software
	Experimentação em Engenharia de Software
	Métodos e Ferramentas de Engenharia de Software
	Desenvolvimento de Software Concorrente
	Desenvolvimento de Software para a Web
	Desenvolvimento de Software para Dispositivos Móveis
	Desenvolvimento de Software para Persistência
	Leitura de Software
	Trabalho Cooperativo Baseado em Computador
	Estimativa de Custo em Projetos de Software
Sistemas operacionais, Redes de computadores, Sistemas Distribuídos e Segurança	Sistemas Operacionais
	Redes e Sistemas Distribuídos
	Segurança

III. Formação Complementar e Humanística

A área de formação complementar é composta por um conjunto de conhecimentos que visam à preparação do egresso para interação com profissionais de outras áreas. Para o Bacharelado em Engenharia de Software destacam-se as matérias que visam dar ao egresso o embasamento sobre o empreendedorismo. A área de formação humanística é composta por um conjunto de conteúdos que objetivam subsidiar a discussão e a compreensão da dimensão humana em relação à Engenharia de Software. As disciplinas optativas-livres também poderão auxiliar na formação complementar e humanística. Prevê-se ainda, a inclusão das disciplinas de Língua Brasileira de Sinais, Relações Étnico-Raciais e Africanidades, Educação Ambiental e Educação em Direitos Humanos. Deste modo, registra-se que o curso ofertará as seguintes disciplinas complementares à formação dos alunos:

Conteúdo	Disciplina
Administração	Empreendedorismo
Direito e Legislação	Ética, Normas e Postura Profissional
Língua Inglesa	Inglês Instrumental I
	Inglês Instrumental II
Educação	Educação Em Direitos Humanos
	Educação Ambiental
	Relações étnico raciais e africanidades
	Língua Brasileira de Sinais

IV. Formação Suplementar

A Formação Suplementar é composta por atividades que permitem a síntese e integração dos conhecimentos adquiridos ao longo do curso, quais sejam:

Conteúdo	Atividades
Trabalho de Conclusão	Trabalho de Conclusão de Curso I e II
Estágio Profissional	Estágio Supervisionado I e II

As unidades curriculares deverão formar o futuro Bacharel em Engenharia de Software para exercer seu papel de profissional e cidadão levando em conta o desempenho de atividades nas áreas de Engenharia de Software e sua responsabilidade social.

Em suma, de acordo com a descrição anterior, as disciplinas foram agrupadas nas seguintes Unidades Curriculares:

- de Matemática
- de Ciência da Computação
- de Engenharia de Software
- de Engenharia de Software Aplicada
- de Tecnologia da Informação
- de Formação Complementar e Humanística
- de Formação Suplementar

De forma mais específica, é possível traçar algumas recomendações em relação aos docentes, de acordo com a área em que atuarão no currículo:

- a) Recomenda-se que os professores que atuarão na Formação Básica em Matemática, Formação Humanística e Formação Complementar tenham formação nas áreas específicas das disciplinas que lecionam. Além disso, é desejável que tenham conhecimentos e experiência profissional que os habilitem a promover a articulação entre os conteúdos desenvolvidos em suas disciplinas e a aplicação em Engenharia de Software;
- b) Recomenda-se que os professores da Formação Básica em Ciência da Computação tenham formação na área de Computação ou Informática. É desejável que estes docentes tenham conhecimentos e experiência profissional que os habilitem a promover a articulação entre os conteúdos desenvolvidos em suas disciplinas e a aplicação em Engenharia de Software;
- c) Os professores da Formação Tecnologia da Informação podem ter formação variada de acordo com a área de aplicação envolvida, sendo geralmente provenientes de Computação e Informática. Além disso, é desejável que disponham de experiência profissional relacionada à aplicação da tecnologia específica em Engenharia de Software;
- d) Os professores das áreas de Formação Básica em Introdução a Engenharia de Software, Formação Tecnológica em Engenharia de Software e Engenharia de Software Aplicada podem ser formadas nas áreas de Computação ou Informática, sendo desejável que tenham cursado a graduação ou pós-graduação em uma destas áreas. Além disso, é desejável que estes docentes tenham experiência profissional e/ou de pesquisa na área da matéria lecionada.

9.2. Disciplinas por departamento

Todas as disciplinas inerentes ao Curso de Bacharelado em Engenharia de Software serão ofertadas no Campus de Russas, em conjunto com os demais cursos ofertados na unidade e levando em conta a demanda de cada semestre. O corpo docente relativo a este curso será constituído por professores qualificados nas áreas de conhecimento adequadas à necessidade de oferta de disciplinas do curso, de acordo com as sugestões reforçadas no item 9.2.

9.3. Ementário das disciplinas

Disciplinas Obrigatórias

Introdução à Computação e a Engenharia de Software

Fundamentos de computação, arquiteturas de computadores e sistemas operacionais. Ambientes de processamento automatizado de informações. Evolução das profissões e características do profissional de sistemas de informação. Conceitos básicos: dado, informação e conhecimento. Operações básicas com números binários. Mudança de base. Computador e seus elementos básicos. Introdução a Sistemas Distribuídos. Conceitos básicos de Engenharia de Software, programas e documentação. SWEBOK e as áreas de conhecimento, Etapas de desenvolvimento de um Software.

Fundamentos de Programação

Algoritmos, Conceitos Fundamentais de Programação, Expressões, Controles de Fluxo, Funções e Procedimentos, Ponteiros, Vetores e Matrizes, Cadeias de Caracteres, Alocação Dinâmica, Tipos Estruturados e Arquivos.

Ética, Normas e Postura Profissional

Noções de ética. Código de ética para engenheiros de software. Visão geral de normas e padrões internacionais pertinentes à engenharia de software, além de leis e resoluções locais. Resolução de conflitos. Como se preparar para reuniões.

Matemática Básica

Teoria dos Conjuntos. Noções de Lógica e Técnicas de Demonstração. Relações. Funções. Números reais. Funções de uma variável real a valores reais. Seqüências e séries. Noções de limite.

Introdução à Processos e Requisitos de Software

Conceitos básicos de Processo de Software. Fases de um Processo de Software e geração de artefatos a cada fase. Tipos de Processo de Software: tradicionais e ágeis. Definição e introdução à elicitação e análise de requisitos de software.

Programação Orienta a Objetos

Paradigma orientado a objetos (abstração, encapsulamento, classes, métodos, objetos, herança, polimorfismo, delegação e outros). Modelagem orientada a objetos usando UML. Noções de princípios de projeto orientado a objetos. Implementação de modelos. Método de desenvolvimento de software orientado a objetos. Visão detalhada de método ágil de desenvolvimento de software.

Desenvolvimento de pequenas aplicações modeladas e implementadas de forma orientada a objetos seguindo um método ágil e o emprego de orientação a objetos.

Arquitetura de Computadores

Sistemas numéricos. Aritmética binária: ponto fixo e ponto flutuante. Organização de computadores: memórias, unidade central de processamento, unidades de entrada e unidades de saída. Linguagens de montagem. Modos de endereçamento, conjunto de instruções. Mecanismos de interrupção e de exceção. Barramento, comunicações, interfaces e periféricos. Organização de memória. Memória auxiliar. Arquiteturas RISC e CISC. Pipeline. Paralelismo de baixa granularidade. Processadores superescalares e superpipeline. Multiprocessadores. Multicomputadores. Arquiteturas paralelas e não convencionais.

Matemática Discreta

Conjuntos. Álgebra dos conjuntos. Relações. Funções. Estruturas algébricas. Reticulados. Álgebra Booleana. Teoria dos Grafos.

Probabilidade e Estatística

Estatística descritiva. Espaço amostral e eventos. Probabilidade e técnicas de contagem. Probabilidade condicional e independência. Variável aleatória, distribuição de probabilidade e momentos. Principais distribuições - binomial, poisson, geométrica, normal, exponencial, gama, erlang, weibull. Conceito e objetivos da estatística. Estatística descritiva. Noções de amostragem. Distribuições amostrais: discreta e contínua. Inferência estatística: teoria da estimação e testes de hipóteses. Regressão linear simples. Correlação. Análise de variância.

Modelagem e Análise de Software

Conceitos fundamentais de Projeto de Software. Análise de Requisitos e principais padrões de Projeto de Software e suas aplicações reais. Considerações em um projeto: compatibilidade, extensibilidade, modularidade, reusabilidade, segurança, etc. Linguagens para Modelagem de Software.

Linguagens de Programação

Tipos (dinâmicos e estáticos). Polimorfismo. Visão geral de paradigmas de programação. Paradigmas de programação (orientado a objetos, orientado a eventos, orientado a aspectos e funcional). Interpretadores. Compiladores. Máquinas virtuais. Básico de linguagens formais (sintaxe, semântica, gramáticas, BNF, diagrama de transição de estados). Análise léxica (scanners). Análise sintática (gramáticas regulares, livres de contexto, LL parsing). Expressões regulares.

Estruturas de Dados

Tipos Abstratos de Dados, Listas Encadeadas, Pilhas, Filas. Árvores, Estruturas Genéricas, Ordenação, Busca e Tabelas de Dispersão.

Sistemas Operacionais

O histórico, o conceito e os tipos de sistemas operacionais. A estrutura de sistemas operacionais. Gerenciamento de memória. Memória virtual. Conceito de processo. Gerência de processador: escalonamento de processos, monoprocessamento e multiprocessamento. Concorrência e sincronização de processos. Alocação de recursos e deadlocks. Gerenciamento de arquivos. Gerenciamento de dispositivos de entrada/saída

Empreendedorismo

Conceito de empreendedorismo. A formação da personalidade. O processo comportamental. Fatores

de sucesso, o perfil do empreendedor. Desenvolvimento de habilidades empreendedoras. Lições e práticas internacionais. Empreendedorismo no Brasil. Importância das MPEs na economia. Globalização dos mercados, dos negócios e das oportunidades. Pesquisas Tecnológicas. Propriedade Intelectual. Transferência de Tecnologia. Papel da inovação. Ambientes de pré-incubação e incubação de idéias. Incubadoras de empresas. Parques Tecnológicos. Capital de Risco. Recursos de Fomento. Fontes de Financiamento. Fundos Setoriais. Programas governamentais. Plano de Negócio. Ferramentas de Plano de Negócios. Projetos.

Redes e Sistemas Distribuídos

Noções de redes de computadores. Introduzir os conceitos fundamentais de sistemas distribuídos, a caracterização de sistemas de computação distribuída, aplicações distribuídas (características e aspectos de projeto), objetivos básicos de sistemas distribuídos (transparência, abertura, escalabilidade e outros). Princípios e aplicações dos principais modelos de sistemas distribuídos: sistemas cliente/servidor e sistemas multi-camadas; sistemas peer-to-peer. Objetos distribuídos: interface x implementação; objetos remotos; chamadas de métodos remotos (RMI). Princípios e uso de serviços de sistemas distribuídos: serviços de nomes; compartilhamento de documentos / recursos distribuídos (ex.: WWW e sistemas de trabalho cooperativo). Princípio e implementação de SOA – Arquiteturas Orientadas a Serviço; Web services. Prática de desenvolvimento de sistemas distribuídos.

Projeto Detalhado do Software

Definição de projeto. Questões fundamentais (persistência dos dados, exceções e outras). Contexto de projeto em vários modelos de desenvolvimento de software (ciclos de vida). Princípios de projeto (encapsulamento de informações, coesão e acoplamento). Interação entre projeto e requisitos. Atributos qualitativos em um projeto (confiabilidade, usabilidade, manutenibilidade, testabilidade, desempenho, segurança, tolerância a falhas e outros). Compromissos (custo-benefício). Relação entre arquitetura de software e projeto de software. Projeto orientado a objetos. Projeto funcional. Noção de projeto baseado em estrutura de dados e projeto orientado a aspectos. Projeto orientado por responsabilidade. Projeto por contratos. Métodos de projeto de software. Padrões de projeto. Reutilização. Projeto de componentes. Projeto de interfaces entre componentes e sistemas. Notações de projeto. Ferramentas de suporte a projeto (análise estática, avaliação dinâmica e outras). Medidas de atributos de projeto (acoplamento, coesão e outras). Métricas de projeto (principais métricas, interpretação).

Interface Humano Computador

Princípios de projeto de interfaces homem-computador. Modos de uso e navegação. Projeto visual (cores, ícones, fontes e outros). Tempo de resposta e retro-alimentação. Elementos de interação (menus, formulários, manipulação direta e outros). Localização e internacionalização. Métodos de projeto de interação. Modelos conceituais e metáforas. Voz, linguagem natural, sons, páginas web. Dispositivos de interação. Heurísticas de avaliação de interfaces. Abordagens para testes realizados com apoio de usuários. Técnicas de testes para páginas web. Visão geral de ferramentas de desenvolvimento de interfaces homem-computador.

Fundamentos de Banco de Dados

Conceitos básicos. Componentes de sistemas de bancos de dados (database systems). Modelagem conceitual (ER e EER). Modelo relacional. Prática de modelagem de dados. Noções de álgebra e cálculo relacional. Mapeamento de esquema conceitual para esquema relacional. Linguagem SQL (extensiva apresentação e prática). Restrições de integridade. Dependências funcionais e formas normais. Transações. Visão geral de mineração de dados e Data Warehousing.

Lógica para Computação

Lógica sentencial e de Primeira ordem. Sistemas dedutivos naturais e axiomáticos. Completeza,

consistência e coerência. Formalização de problemas. Lógicas Temporais para a Validação de Sistemas.

Processo de Software

Conceitos e terminologia. Infraestrutura de processos (pessoas, ferramentas, treinamentos e outros). Modelagem e especificação de processos de software. Medição e análise de processos de software. Melhoria de processos de software (individual e equipe). Análise e controle de qualidade (prevenção de defeitos, revisão de processos, métricas de qualidade, análise de causa e outros). Níveis de definição de processos. Modelos de ciclo de vida (ágil, processos “pesados”, cascata, espiral, modelo V e outros). Modelos de processos e padrões (IEEE, ISO e outros). Modelo, definição, medida, análise e melhoria tanto de processo de software individual quanto de equipe. Personalização de processo. Requisitos para processos de software (ISO/IEEE 12207). Visão geral do CMMI e ITIL. Detalhada apresentação do MSP.BR (guias). Implementação do MPS.BR.

Requisitos de Software

Princípios de modelagem como decomposição e abstração. Pré e pós condições. Invariantes. Visão geral de modelos matemáticos e linguagens de especificação como Z, VDM, NFR e GORE. Interpretação de modelos (sintaxe e semântica). Modelagem de informações (modelo entidade-relacionamento e diagrama de classes). Modelagem de comportamento (diagramas de estados, diagramas de casos de uso, diagramas de interação). Modelagem de estrutura (arquitetura). Modelagem de domínio. Modelagem funcional. Modelagem de processos de negócios. Padrões de análise. Fundamentos como completitude, consistência, robustez, análise estática, simulação, verificação de modelos, segurança, safety, usabilidade, desempenho, análise de causa/efeito, priorização, análise de impacto, rastreabilidade. Definição de requisitos de produto, projeto, restrições, fronteiras de um sistema. Processo de requisitos. Níveis de requisitos (necessidades, objetivos, requisitos dos usuários, requisitos de sistema, requisitos de software. Características de requisitos (testáveis, verificáveis e outras). Gerência de requisitos. Interação entre requisitos e arquitetura. Fontes e técnicas de elicitação. Documentação de requisitos (normas, tipos, audiência, estrutura, qualidade). Especificação de requisitos. Revisões e inspeções. Construção de protótipos para validar requisitos. Relação com testes de aceitação.

Projeto e Análise de Algoritmos

Desenvolvimento de algoritmos. Técnicas de projeto de algoritmos eficientes. Análise de Algoritmos (Tempo de Processamento e Operações Elementares), Técnicas de prova de cotas inferiores. Exemplos de análise de algoritmos iterativos e recursivos. Algoritmos e estruturas de dados para problemas em grafos, Programação dinâmica. Algoritmos probabilísticos. Teoria da Complexidade, Problemas de Decisão, Transformações Polinomiais, Classes de problemas: P, NP, Co-NP, P-SPACE.

Manutenção de Software

Conceitos e terminologia. Categorias (tipos) de manutenção. Questões técnicas e gerenciais de manutenção. Estimativa de custo de manutenção. Métricas/medidas para manutenção. Processos e atividades de manutenção. Compreensão de programas. Reengenharia. Engenharia reversa. Norma IEEE Std 14764-2006. Refatoração. Transformação de programas.

Gerência de Projetos

Conceitos, terminologia e contexto de gerência de projetos. Ciclo de vida de produto e projeto. Interessados (stakeholders). Organização de empresas (funcionais, matriciais e baseadas em projetos). Estratégias para seleção de projetos. Processos de gerência de projetos. Gerência de escopo. Gerência de tempo (definição de atividades, seqüenciamento de atividades, estimativa de recursos, estimativa de duração, desenvolvimento de cronograma e controle de cronograma). Gerência de custos (estimativas, orçamento e controle). Gerência de qualidade. Gerência de recursos humanos. Gerência de comunicação. Gerência de riscos. Gerência de aquisições. Gerência de integração (desenvolver carta

de projeto, desenvolver escopo preliminar, desenvolver plano de gerência de projeto, dirigir e gerenciar a execução de projetos, monitorar e controlar atividades de projeto, controle de mudanças e fechamento do projeto). Estabelecer relações com o MPS.BR. Gerência de aquisições deve ser observada da perspectiva do Guia de Aquisições de Software e Serviços Correlatos (MPS.BR).

Qualidade de Software

Definições e terminologia de qualidade de software. Custos e impactos de baixa qualidade. Custo de um modelo de qualidade. Terminologia para características de qualidade de software (ISO 9126-1). Papel de pessoas, processos, métodos, ferramentas e tecnologias em qualidade. Padrões de qualidade (ISO 9001, ISO 9003-04, IEEE Std 1028-2008, IEEE Std 1465-2004, IEEE Std 12207-2008, ITIL). Revisões, auditoria e inspeções. Modelos e métricas de qualidade de software. Aspectos relacionados a qualidade de modelos de processos de software. Visão geral do CMMI. MPS.BR. Planejamento de qualidade. Garantia da qualidade. Análise de causa e prevenção de defeitos. Avaliação de atributos de qualidade. Métricas e medidas de qualidade de software. Desenvolver planos de qualidade de software em conformidade com o padrão IEEE Std 730-2002.

Verificação e Validação

Objetivos e restrições de V&V (Verificação e Validação). Planejamento de V&V. Documentação de estratégias de V&V, testes e outros artefatos. Medidas e Métricas. Análise estática de código. Atividades de V&V ao longo do ciclo de vida de um produto. Revisão de software. Testes de unidade. Análise de cobertura. Técnicas de teste funcional (caixa preta). Testes de integração. Desenvolvimento de casos de teste baseados em casos de uso e histórias de usuários. Testes de sistema. Testes de aceitação. Testes de atributos de qualidade. Testes de regressão. Ferramentas de teste (combinação com ferramentas de integração contínua). Análise de relatórios de falha. Técnicas para isolamento e falhas (depuração). Análise de defeitos. Acompanhamento de problemas (tracking). IEEE Std 1012-2004.

Disciplinas Optativas

Trabalho Cooperativo Baseado em Computador

Colaboração e Cooperação. Sociedade do conhecimento. Modelos de gestão e organização baseados em conhecimento. Organizações de aprendizagem. Modelos para ambientes de trabalho cooperativo baseado em computador. Tecnologias de suporte à comunicação e cooperação. Planejamento de processos organizacionais cooperativos.

Especificação Formal de Software

Estudo de métodos formais para desenvolvimento de software. Importância da Especificação Formal na Engenharia de Software. Uso de Ferramentas de apoio ao desenvolvimento formal.

Arquitetura de Software

Definição de arquitetura de software. Importância e impacto em um software. Estilos arquiteturais (pipeand-filter, camadas, transações, publish-subscribe, baseado em eventos, cliente-servidor e outros). Relação custo/benefício entre vários atributos. Questões de hardware em projeto de software. Rastreabilidade de requisitos e arquitetura de software. Arquiteturas específicas de um domínio e linhas de produto. Notações arquiteturais (visões, representações, diagramas de componentes e outros). Reutilização.

Leitura de Software

Estudar, investigar, analisar e discutir projetos de softwares existentes e “vencedores”.

Compiladores

Introdução a Compiladores, Análise Léxica, Análise Sintática, Análise Semântica, Geração de Código, Tópicos Especiais em Compiladores.

Integração de Aplicações

Definição de integração de aplicações. Desafios de integração. Abordagens de integração (transferência de arquivos, bases de dados compartilhadas, chamada de procedimento remoto e troca de mensagens). Padrões para integração de aplicações.

Desenvolvimento de Software Concorrente

Liveness. Safety. Semáforos. Locks. Threads. Deadlocks. Implementações de algoritmos concorrentes. Linguagem MPI.

Desenvolvimento de Software para Dispositivos Móveis

Visão geral sobre dispositivos móveis: Comparação entre dispositivos móveis e computadores convencionais; A linguagem Java para o desenvolvimento de aplicações para dispositivos móveis: JME; Arquitetura JME: Configurações (CLDC, CDC, etc) e Perfis (MIDP, Personal Profile, etc); Recursos da linguagem: MIDlets (aplicação / interface), GCF (framework de comunicação), RMS (framework de dados), High Level API e Low Level API (canvas). Frameworks e Bibliotecas para desenvolvimento de aplicações móveis.

Desenvolvimento de Software para Persistência

Definição de persistência. Persistência empregando arquivos binários, documentos XML, objetos serializáveis, SGBDs. Tecnologias para persistência de informações. Persistência de objetos usando base relacional.

Métodos e Ferramentas de Engenharia de Software

Métodos heurísticos, formais e de construção de protótipos. Ferramentas para auxiliar na produção de requisitos, projeto, construção, testes e manutenção. Ferramentas de gerência de configuração, gerência de projeto, processo de software, qualidade e outras.

Desenvolvimento de Software para Web

Conceitos básicos de Internet e Web, histórico e a W3C. Linguagem de marcação HTML, folhas de estilo, introdução a Javascript e Web 2.0. Construção de páginas dinâmicas: principais servidores web, servidor Tomcat, servlets, jsp, scriptlets e interação com banco de dados. Introdução aos principais Frameworks para desenvolvimento Web.

Experimentação em Engenharia de Software

Conceituação e esclarecimento acerca de experimento controlado, estudos de caso e surveys. Processo de desenvolvimento de um projeto de pesquisa (inclui atividades, formulação de questões, construção de teoria e análise qualitativa/quantitativa de dados). Investigação de experimentos científicos em engenharia de software. Prática acompanhada de pequeno experimento em engenharia de software.

Reuso de Software

Conceitos básicos e importância de Reuso de Software. Principais abordagens de reuso: Engenharia de Domínio, Linha de Produtos de Software, Desenvolvimento Baseado em Componentes, Arquitetura

Orientada a Serviços, Padrões e Frameworks.

Gerência de Configuração

Conceitos e terminologia. Processos de gerência de configuração. Identificação de itens de configuração. Atributos a serem registrados para cada item de configuração. Armazenamento. Controle de mudanças. Relatórios de status. Controle de versões e linhas base ou de referência (baselines). Gerência de configuração segundo o MPS.BR. Papéis em gerência de configuração. Normas (IEEE 828). Princípios de gerência de configuração e relação com atividades de desenvolvimento de software. Gerência de configuração segundo desenvolvimento ágil, técnica de builds frequentes e desenvolvimento iterativo. Gerência de configuração para diferentes tipos de produtos (compostos, multiplataforma, múltiplas variantes, críticos, pequenos, médios e grandes). Gerência de configuração para desenvolvimento de software distribuído geograficamente, múltiplos interessados e desenvolvimento paralelo. Melhoria de gerência de configuração. Considerações práticas acerca de gerência de configuração de software. Ferramentas.

Estimativa de Custo em Projetos de Software

Estimativa do tamanho do software a ser desenvolvido; Estimativa dos riscos e incertezas do projeto; Estimativa de esforço necessário para construir o software; Estimativa de prazo necessário para construir o software e Estimativa do custo do projeto. Modelos de Estimativas como: Estimativa em pontos de Casos de Uso e Análise de Pontos de Função, COCOMO e COCOMO II.

Segurança

Ameaças. Segurança como atributo qualitativo de projeto de software. Autenticação. Autorização. Integridade. Confidencialidade. Criptografia (chaves simétricas e assimétricas). Infraestrutura de chaves públicas brasileiras (ICP-Brasil). Certificados digitais. Assinaturas digitais. Desenvolvimento de software seguro. Noções de auditoria de sistemas. Norma NBR 27002.

Teoria da Computação

Modelos computacionais universais. Computabilidade. Funções recursivas. Introdução à complexidade de problemas e tópicos avançados.

Linguagens Formais

Introdução. Linguagens, gramáticas e expressões regulares, autômatos finitos. Linguagens e gramáticas livre-do-contexto e autômatos de pilha. Linguagens sensíveis ao contexto. Hierarquia de classes de linguagens. Tópicos especiais e aplicações das linguagens formais e autômatos

Inglês Instrumental I

Vocabulário ligado à Informática. Técnicas de Leitura e Compreensão. Tradutores. Verbos e tempos verbais. Nomes e Pronomes.

Inglês Instrumental II

Conversação, leitura de artigos e jornais da área. Escrita de trabalhos técnicos. Apresentação de seminários. Noções avançadas de gramática e compreensão de texto.

Inteligência Artificial

Conceito de IA, Histórico e Metas. Agentes Inteligentes. Solução de problemas, busca e jogos. Sistemas Lógicos, Conhecimento e Raciocínio. Sistemas Baseados em Conhecimento. Planejamento. Incerteza, Probabilidade e Teoria da Decisão. Aprendizado. Linguagem e Comunicação. Percepção.

Relações Étnico-Raciais e Africanidades

EMENTA: Negritude e pertencimento étnico. Conceitos de africanidades e afrodescendência. Cosmovisão africana: valores civilizatórios africanos presentes na cultura brasileira. Ancestralidade e ensinamentos das religiosidades tradicionais africanas nas diversas dimensões do conhecimento no Brasil. Introdução à geografia e história da África. As origens africanas e as nações africanas representadas no Brasil. O sistema escravista no Brasil e no Ceará. Aportes dos africanos à formação social e cultural do Brasil e do Ceará. Personalidades africanas, afrodescendentes e da diáspora negra que se destacaram em diferentes áreas do conhecimento. Contexto das Ações Afirmativas hoje. Atualização do legado africano no Brasil. Desconstrução de preconceitos e desdobramentos teórico-práticos para a atuação do profissional na sua área de inserção no mercado de trabalho.

Educação Ambiental

EMENTA: Educação Ambiental, conceitos e metodologias na pesquisa e no ensino. Princípios da Educação Ambiental. Fundamentos filosóficos e sociológicos da Educação Ambiental. Tratado de Educação Ambiental para Sociedades Sustentáveis; A Agenda XXI; A Carta da Terra e outros marcos legais da EA. Educação Ambiental e sua Contextualização (Urbana e Rural). Paradigmas Epistemológico-educativos Emergentes e a Dimensão Ambiental. Educação Ambiental: uma abordagem crítica. Educação Ambiental Dialógica e a Práxis em Educação Ambiental.

Educação em Direitos Humanos

EMENTA: Direitos Humanos, democratização da sociedade, cultura e paz e cidadanias. O nascituro, a criança e o adolescente como sujeitos de direito: perspectiva histórica e legal. O ECA e a rede de proteção integral. Educação em direitos humanos na escola: princípios orientadores e metodologias. O direito à educação como direito humano potencializador de outros direitos. Movimentos, instituições e redes em defesa do direito à educação. Igualdade e diversidade: direitos sexuais, diversidade religiosa e diversidade étnica. Os direitos humanos de crianças e de adolescentes nos meios de comunicação e nas mídias digitais.

9.4. Estágio Supervisionado

O estágio visa assegurar o contato do formando com situações, contextos e instituições, permitindo que conhecimentos, habilidades e atitudes se concretizem em ações profissionais, promovendo a articulação entre teoria e prática, contribuindo para a consolidação das competências desenvolvidas ao longo do curso, tendo em vista o perfil de profissional que se deseja formar e pressupõe supervisão sistemática, feita conjuntamente por professor supervisor e por profissional do campo, com base em planos de estágio elaborados em conjunto pelas unidades de ensino e organizações que oferecem estágio.

Oportunidades de estágios deverão ser oferecidas aos alunos do curso de Engenharia de Software dentro da UFC nas áreas de abrangência do curso, através do engajamento nos projetos em andamento. Também será incentivada a participação dos acadêmicos em estágios fora da Instituição, visando o aperfeiçoamento e a diversificação no desenvolvimento de suas atividades. Para tanto, será necessário que a UFC firme convênios de parcerias com empresas.

O Estágio Supervisionado será acompanhado pelo Professor-Supervisor de Estágio e as experiências individuais serão relatadas pelo estudante estagiário. Ao final do Estágio, serão avaliadas as experiências que poderão ser sistematizadas para publicações e apresentação em eventos relativos à Engenharia de Software. A carga horária do estágio será de 320 horas, distribuída nos dois últimos semestres do curso.

9.5. Trabalho de Conclusão de Curso

O Trabalho de Conclusão de Curso (TCC) objetiva capacitar os alunos na elaboração e apresentação de um trabalho acadêmico produzido a partir da inter-relação de conhecimentos vistos durante o curso, seguindo normas técnicas relativas à formatação do documento, às referências bibliográficas e às citações, podendo ser desenvolvido de diferentes formas, monografia, produto, projeto, artigo científico.

O Curso de Bacharelado em Engenharia de Software possui um TCC, que envolverá todos os procedimentos de uma investigação técnico-científica, a ser desenvolvida pelo estudante ao longo dos dois últimos semestres do curso, compreendendo a carga horária de 128 horas. Procura-se a aplicação do método científico nos trabalhos TCC, que tem como objetivo principal a aplicação das tecnologias abordadas nas disciplinas específicas da Engenharia de Software. Considera-se sempre a evolução da área nas atividades. O trabalho poderá ser desenvolvido nas diversas áreas de pesquisa em Engenharia de Software, tais como gerência de projetos, qualidade de software, reuso de software, entre outros temas que podem ser acordados com o professor orientador dessa atividade. O trabalho deverá ser entregue em formato acadêmico e defendido perante uma banca de três professores da área, de acordo com os critérios gerais da UFC.

A regulamentação nas instâncias competentes possibilitará a exposição dos critérios adequados a cada forma prevista.

9.6. Atividades Complementares

As atividades complementares possibilitam o reconhecimento de habilidades e competências do aluno, inclusive adquiridas fora do ambiente escolar, contribuindo para a flexibilização do currículo com a contabilização no histórico escolar de vivências adquiridas fora da sala de aula. Trata-se, portanto, de componentes curriculares enriquecedores da formação.

Na UFC, essas atividades encontram-se normatizadas pela Resolução CEPE nº. 7/2005, segundo a qual:

Art. 2o. – São consideradas atividades complementares:

I – Atividades de iniciação à docência;

II – Atividades de iniciação à pesquisa;

III – Atividades de extensão;

IV – Atividades artístico-culturais e esportivas;

V – Atividades de participação e/ou organização de eventos;

VI – Experiências ligadas à formação profissional e/ou correlatas;

VII – Produção Técnica e/ou Científica;

VIII – Vivências de gestão;

IX – Outras atividades, estabelecidas de acordo com o Art. 3o. desta Resolução.

Art. 3º. – As Coordenações de Cursos de Graduação poderão aprovar normatizações específicas, incluindo estratégias pedagógico-didáticas não previstas no Art. 2o. desta Resolução e estipulando carga horária mínima integralizada ou período cursado das Atividades Complementares.

É importante registrar que a regulamentação deste componente trará a discriminação de horas por atividades e os modos de acompanhamento deste componente curricular

10. Integralização Curricular

O currículo do Curso de Bacharelado em Engenharia de Software é composto por disciplinas que têm enfoque nas práticas mais comuns de engenharia de software e que são mais utilizadas para resolução das necessidades apresentadas pelo mercado. Composto por um conjunto de disciplinas a somar 3.072 horas/aula, este currículo compreende disciplinas a serem estudadas com objetivo da formação de um profissional qualificado e em sintonia com o mercado. Uma vez que o perfil profissional desejado ao egresso do curso que propomos é envolver, entre outras, a capacidade de desenvolver implementar e utilizar soluções de TIC, requeremos que o aluno demonstre a evolução de seus conhecimentos através da execução do TCC (Trabalho de Conclusão de Curso) e do Estágio Supervisionado.

Na próxima página, apresentamos a integralização curricular do curso de Engenharia de Software:

Integralização Curricular do Curso de Engenharia de Software – Campus Russas

Componentes Curriculares Obrigatórios					
Semestre	Nome do Componente Curricular	Carga Horária Teórica	Carga Horária Prática	Carga Horária Total	Pré-requisito(s)
1º	Fundamentos de Programação	48 h	48 h	96 h	-
1º	Ética, Normas e Postura Profissional	64 h	0 h	64 h	-
1º	Introdução à Computação e Engenharia de Software	96 h	0 h	96 h	-
1º	Matemática Básica	64 h	0 h	64 h	-
2º	Arquitetura de Computadores	64 h	0 h	64 h	-
2º	Programação Orientada a Objetos	32 h	32 h	64 h	Fundamentos de Programação
2º	Matemática Discreta	64 h	0 h	64 h	Matemática Básica
2º	Probabilidade e Estatística	64 h	0 h	64 h	Matemática Básica
2º	Introdução a Processos e Requisitos de Software	64 h	0 h	64 h	Introdução à Computação e Engenharia de Software
3º	Estrutura de Dados	32 h	32 h	64 h	Fundamentos de Programação
3º	Sistemas Operacionais	48 h	16 h	64 h	Arquitetura de Computadores
3º	Análise e Projeto de Sistemas	64 h	0 h	64 h	Introdução a Processos e Requisitos de Software
3º	Linguagens de Programação	64 h	0 h	64 h	Fundamentos de Programação
3º	Empreendedorismo	64 h	0 h	64 h	-
4º	Fundamentos de Banco de Dados	64 h	0 h	64 h	-

4°	Lógica para Computação	64 h	0 h	64 h	Matemática Básica
4°	Interface Humano-Computador	64 h	0 h	64 h	-
4°	Projeto Detalhado de Software	64 h	0 h	64 h	Análise e Projeto de Sistemas
4°	Redes e Sistemas Distribuídos	64 h	0 h	64 h	-
5°	Gerência de Projetos de Software	64 h	0 h	64 h	-
5°	Projeto e Análise de Algoritmos	64 h	0 h	64 h	Estrutura de Dados
5°	Processos de Software	64 h	0 h	64 h	Introdução a Processos e Requisitos de Software
5°	Requisitos de Software	64 h	0 h	64 h	Introdução a Processos e Requisitos de Software
6°	Manutenção de Software	64 h	0 h	64 h	Análise e Projeto de Sistemas
6°	Qualidade de Software	64 h	0 h	64 h	-
6°	Verificação e Validação	64 h	0 h	64 h	-
7°	Estágio Supervisionado I	0 h	160 h	160 h	-
7°	Projeto de Pesquisa Científica e Tecnológica	16 h	16 h	32 h	Gerência de Projetos de Software e Qualidade de Software
7°	Trabalho de Conclusão de Curso I	32 h	0 h	32 h	-
8°	Arquitetura de Software	32 h	32 h	64 h	Projeto Detalhado de Software
8°	Estágio Supervisionado II	0 h	160 h	160 h	Estágio Supervisionado I
8°	Trabalho de Conclusão de Curso II	0 h	96 h	96 h	Trabalho de Conclusão de Curso I
Componentes Curriculares Optativos					
Desenvolvimento de Software para Web		64 h	0 h	64 h	Programação Orientada a Objetos

Compiladores	64 h	0 h	64 h	Linguagens de Programação
Inteligência Artificial	64 h	0 h	64 h	Lógica para Computação
Linguagens Formais e Autômatos	64 h	0 h	64 h	-
Teoria da Computação	64 h	0 h	64 h	Matemática Discreta
Trabalho Cooperativo Baseado em Computadores	64 h	0 h	64 h	-
Tópicos Especiais I	64 h	0 h	64 h	-
Tópicos Especiais II	64 h	0 h	64 h	-
Especificação Formal de Software	64 h	0 h	64 h	-
Gerência de Configuração	64 h	0 h	64 h	-
Leitura de Software	64 h	0 h	64 h	-
Reuso de Software	64 h	0 h	64 h	-
Segurança	64 h	0 h	64 h	-
Estimativa de Custos em Projetos de Software	32 h	32 h	64 h	Gerência de Projetos de Software
Integração de Aplicações	32 h	32 h	64 h	-
Experimentação em Engenharia de Software	32 h	32 h	64 h	-
Desenvolvimento de Software Concorrente	32 h	32 h	64 h	-
Redes Sociais	48 h	16 h	64 h	Projeto Detalhado de Software
Sistemas Multiagentes	32 h	32 h	64 h	Programação Orientada a Objetos e Sistemas Operacionais
Métodos e Ferramentas da Engenharia de Software	16 h	48 h	64 h	Verificação e Validação

Introdução ao Desenvolvimento de Jogos	32 h	32 h	64 h	Projeto Detalhado de Software
Computação em Nuvem	32 h	32 h	64 h	Redes e Sistemas Distribuídos e Fundamentos de Banco de Dados
Desenvolvimento de Software para Persistência	16 h	48 h	64 h	Fundamentos de Banco de Dados e Programação Orientada a Objetos
Desenvolvimento de Software para Dispositivos Móveis	48 h	16 h	64 h	Redes e Sistemas Distribuídos e Programação Orientada a Objetos

11. Acompanhamento e Avaliação do Projeto Pedagógico

Esta avaliação terá como objetivo possibilitar a retroalimentação do processo de elaboração e implementação do projeto para que seja possível detectar os pontos a serem revistos, ajustados e reformulados. Parte-se do entendimento do projeto pedagógico como um processo dinâmico, aberto e flexível que se constrói continuamente com a participação de toda a comunidade acadêmica diretamente relacionada ao curso (docentes, discentes e técnico-administrativos), bem como com a colaboração de representantes da sociedade civil, com o intuito de manter o curso sintonizado com as necessidades do ambiente externo e propiciar o aperfeiçoamento constante das suas condições de ensino.

Esse esforço de construção coletiva implica a seleção de valores e conhecimentos. Para tal, é necessário a construção de um espaço democrático de tomada de decisões, que estimule o diálogo constante entre os participantes envolvidos, procurando construir uma dinâmica no cotidiano educativo e, conseqüentemente, assumir o compromisso com um pacto pedagógico (Veiga, 1998).

Essa participação é fundamental, pois determina a legitimidade do projeto, mas não deve ser imposta; deve ser conquistada por uma equipe coordenadora (Veiga, 1998), pois a imposição só gera projetos burocráticos que se revelam ineficientes (Gadotti, 1997).

Por outro lado, ainda de acordo com Veiga (2003), a compreensão do projeto como processo inscreve-o numa inovação emancipatória ou edificante, que enfatiza o desenrolar da construção e reconstrução do projeto, pautada pelo debate entre os atores envolvidos sobre a realidade interna da instituição e o contexto social mais amplo. Nesse caso, o desenvolvimento do projeto dá-se: (i) pela via democrática; (ii) de dentro para fora; (iii) numa perspectiva globalizante e sistêmica; (iv) sem separação entre fins e meios, uma vez que a ação incide sobre ambos; (v) e pressupõe uma ruptura que, acima de tudo, predisponha as pessoas e a instituição para a indagação e para a emancipação.

Assumir essa perspectiva, implica conceber que o PPC não se restringe a um programa de estudos, a um agrupamento de planos de ensino ou a um conjunto de atividades ordenadas; implica reconhecer que o PPC não é algo estático, um documento, que uma vez construído deve ser arquivado ou enviado para as instâncias competentes como prova do cumprimento de formalidades burocráticas (Veiga, 1995). Contrariamente, a (re)construção de um projeto educativo decorre continuamente ao longo de um processo, pois o projeto não é um produto fechado. O seu processo de construção traduz-se nas tarefas de pensar/elaborar o projeto, pensar/realizar o projeto, pensar/avaliar o projeto e pensar/reformular o projeto, pois o que se pretende vai além da reorganização da educação, aquilo que se busca é a melhoria da qualidade de todo o processo vivenciado.

Para efetivação dessa avaliação poderão se instrumentos e técnicas diversas, tais como questionários, entrevistas, grupos focais, entre outras metodologias que permitam

o levantamento de dados acerca da implementação do curso de Engenharia de Software do *Campus* de Russas.

Como estratégias de ação, planeja-se levar a cabo:

- A discussão ampla do projeto pedagógico com o corpo docente do curso para avaliação da proposta formativa, buscando averiguar sua adequação aos parâmetros curriculares da área das TIC, bem como sua relação com o contexto local e regional em que o curso está inserido e com a qualificação e experiência acadêmica e profissional de seus professores. Entende-se que o colegiado do Curso será proponente e executor dessa conjectura, desenvolvendo adequadamente os seus instrumentos e metodologias. Contudo, pode-se adiantar, sabe-se que a necessidade de avaliação do projeto pedagógico é permanente, então se subentende que semestralmente o ensino, o currículo e o PPC deverão ser objeto de discussão, refletindo sobre o prescrito nesta projeção e o exequível.
- A escuta dos alunos, semestralmente, no decorrer da instalação do curso, para averiguar se suas expectativas em relação à formação estão sendo atendidas, para levantar as possíveis dificuldades existentes nas disciplinas, nos processos de ensino e de aprendizagem, como também se as condições de infraestrutura (salas de aula, laboratório, acervo da biblioteca) atendem as suas necessidades.
- A promoção de encontros e debates, anuais, incluindo representantes da sociedade sejam lideranças comunitárias, membros de associações profissionais, empresários, entre outros, para avaliar se o curso vai ao encontro das demandas sociais e econômicas.

Por fim, sobre o acompanhamento e avaliação do Projeto Pedagógico deste curso, expõe-se o entendimento da necessidade de que o acompanhamento e a avaliação desta projeção sejam feitos por todos os membros da comunidade acadêmica, e com a mesma regularidade. Assim, dizemos que professores e estudantes semestralmente farão avaliação da proposta, analisando sua concepção, sua execução e o atendimento aos objetivos expostos neste documento. Pretende-se seguir as recomendações da CPA, analisando: o planejamento docente, a atuação do professor na execução do planejado, as formas de acompanhamento da aprendizagem discente. Do mesmo modo, entende-se a importância de que professores e estudantes se auto-avaliem, identificando-se, possivelmente, procedimentos, hábitos e métodos capazes de progressão do ensino e obtenção de qualidade na formação profissional dos estudantes.

12. Acompanhamento e Avaliação dos Processos de Ensino e Aprendizagem

De modo geral, os mecanismos de avaliação da aprendizagem do aluno em sala são muito particulares a cada professor que os determina no momento da elaboração do plano de ensino. Entretanto, o curso de Engenharia de Software de Russas pretende

incentivar a definição conjunta entre docentes e discentes das formas de avaliação e a utilização de instrumentos diversos, que além das provas objetivas, possam contemplar a realização de seminários, a elaboração de relatórios, a construção de projetos, protótipos, entendendo que a aprendizagem não se dá através da simples memorização de conteúdos, mas sim, a partir da sua compreensão e contextualização. Ao lado disso, pretende-se a cada início de ano, realizar alguns fóruns de avaliação dos resultados do desempenho dos alunos em relação aos objetivos de cada disciplina e atividade, a fim de detectar dificuldades na aprendizagem, replanejar e tomar decisões que diminuam o repesamento e evasão de alunos.

Institucionalmente, a avaliação dos processos de ensino e de aprendizagem seguirá as normas estabelecidas pelo Regimento Geral da UFC, segundo o qual, citamos:

Art. 109. A avaliação do rendimento escolar será feita por disciplina e, quando se fizer necessário, na perspectiva de todo o curso, abrangendo sempre a assiduidade e a eficiência, ambas eliminatórias por si mesmas.

§ 1o Entende-se por assiduidade a frequência às atividades correspondentes a cada disciplina.

§ 2o Entende-se por eficiência o grau de aproveitamento do aluno nos estudos desenvolvidos em cada disciplina.

Art. 110. A verificação da eficiência em cada disciplina será realizada progressivamente durante o período letivo e, ao final deste, de forma individual ou coletiva, utilizando formas e instrumentos de avaliação indicados no plano de ensino e aprovados pelo Departamento.

§ 1o As avaliações escritas, após corrigidas, e suas notas transcritas nos mapas de notas pelo professor, serão devolvidas ao aluno.

§ 2o A devolução de que trata o parágrafo anterior deverá fazer-se pelo menos até 07 (sete) dias antes da verificação seguinte.

§ 3o Será assegurada ao aluno a segunda chamada das provas, desde que solicitada, por escrito, até 03 (três) dias úteis decorridos após a realização da prova em primeira chamada.

§ 4o É facultado ao aluno, dentro de 03 (três) dias úteis após o conhecimento do resultado da avaliação, solicitar justificadamente a respectiva revisão pelo próprio docente, encaminhando o pedido através do chefe do Departamento correspondente.

Art. 111. Os resultados das verificações do rendimento serão expressos em notas na escala de 0 (zero) a 10 (dez), com, no máximo, uma casa decimal.

Art. 112. A verificação da eficiência compreenderá as avaliações progressivas e a avaliação final.

§ 1o Entende-se por avaliações progressivas, aquelas feitas ao longo do período letivo, num mínimo de duas, objetivando verificar o rendimento do aluno em relação ao conteúdo ministrado durante o período.

§ 2o Entende-se por avaliação final, aquela feita através de uma verificação realizada após o cumprimento de pelo menos 90% (noventa por cento) do conteúdo programado para a disciplina no respectivo período letivo.

Art. 113. Na verificação da assiduidade, será aprovado o aluno que frequentar 75% (setenta e cinco por cento) ou mais da carga horária da disciplina, vedado o abono de faltas.

Art. 114. Na verificação da eficiência, será aprovado por média o aluno que, em cada disciplina, apresentar média aritmética das notas resultantes das avaliações progressivas igual ou superior a 07 (sete).

§ 1o O aluno que apresentar a média de que trata o *caput* deste artigo, igual ou superior a 04 (quatro) e inferior a 07 (sete), será submetido à avaliação final.

§ 2o O aluno que se enquadrar na situação descrita no parágrafo anterior será aprovado quando obtiver nota igual ou superior a 04 (quatro) na avaliação final, média final igual ou superior a 05 (cinco), calculada pela seguinte fórmula:

$$MF = \frac{NAF + \sum NAP/n}{2}$$

onde: MF = Média Final;

NAF = Nota de Avaliação Final;

NAP = Nota de Avaliação Progressiva;

n = Número de Avaliações Progressivas.

§ 3º Será reprovado o aluno que não preencher as condições estipuladas no art. 113, no caput e § 2º do art. 114.

Art. 115. Constará da síntese de rendimento escolar o resultado final de aprovação do aluno, expresso por:

- a) Média aritmética das avaliações progressivas;
- b) nota de avaliação final;
- c) média final;
- d) frequência.

Referências

BRASSCOM Brasscom. URL: <http://brasscom.com.br>.

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). URL: <http://www.cnpq.br>.

FINEP Financiadora de Estudos e Projetos (FINEP). URL: <http://www.finep.gov.br>.

IEEE 610.12-1990 IEEE STD 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, IEEE Computer Society, 1990.

PORTO, B. S.; RODRIGUES, Y. K. *Sugestão de Sumário para Projeto Pedagógico de Curso de Graduação*. Pró-Reitoria de Graduação, Universidade Federal do Ceará, 2013.

PMBOK 2008 A Guide to the Project Management Body of Knowledge, PMBOK® Guide, Fourth Edition, ANSI/PMI 99-001-2008, 2008.

SEEK 2004 SEEK 2004 Software Engineering 2004 — Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, 2004. Disponível em <http://sites.computer.org/ccse/>.

Sommerville Software Engineering, 8th edition, Ian Sommerville, Pearson Addison-Wesley, 2006.

SWEBOK 2004 Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, 2004. Disponível em <http://swebok.org>.

Softex Softex: Excelência em Software. URL: <http://www.softex.br>.

UNIVERSIDADE FEDERAL DO CEARÁ. *Regimento Geral*. Fortaleza, 2013.

VEIGA, I. P. A. Projeto político pedagógico da escola: uma construção coletiva. In: Veiga, I. P. A. (Org.) *Projeto político pedagógico da escola: uma construção possível*. Campinas, SP: Papyrus, 1995.

VEIGA, I. P. A. Perspectivas para reflexão em torno do projeto político-pedagógico. In: VEIGA, I. P. A. (Org.). *Escola: espaço do projeto político-pedagógico* (7a ed.). Campinas, SP: Papyrus, 1998.

VEIGA, I. P. A. Inovações e projeto político-pedagógico: uma relação regulatória ou emancipatória? In: *Caderno Cedes*, 23 (61), 267-281, 2003.